UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/037,683	01/04/2002	Robert S. Brayton	200302369-1	8294

22879 7590 07/22/2010

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
3404 E. Harmony Road
Mail Stop 35
FORT COLLINS, CO 80528

EXAMINER	
BASEHOAR, ADAM L	
ART UNIT	**PAPER NUMBER**
2178	

NOTIFICATION DATE	DELIVERY MODE
07/22/2010	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

JERRY.SHORMA@HP.COM
ipa.mail@hp.com
laura.m.clark@hp.com

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/037,683
Filing Date: January 04, 2002
Appellant(s): BRAYTON ET AL.

Dan C. Hu
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 04/30/10 appealing from the Office action mailed 12/02/09.

(1) Real Party in Interest

A statement identifying by name the real party of interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The following is a list of claims that are rejected and pending in the application:

Claims 29-47 have been finally rejected and are the subject of the submitted Appeal.

(4) Status of Amendments After Final

The examiner has no comment on the appellant's statement of the status of amendments after final rejection contained in the brief.

(5) Summary of Claimed Subject Matter

The examiner has no comment on the summary of claimed subject matter contained in the brief.

(6) Grounds of Rejection to be Reviewed on Appeal

The examiner has no comment on the appellant's statement of the grounds of rejection to be reviewed on appeal. Every ground of rejection set forth in the Office action from which the appeal is taken (as modified by any advisory actions) is being maintained by the examiner except for the grounds of rejection (if any) listed under the subheading "WITHDRAWN REJECTIONS." New grounds of rejection (if any) are provided under the subheading "NEW GROUNDS OF REJECTION."

(7) Claims Appendix

The examiner has no comment on the copy of the appealed claims contained in the Appendix to the appellant's brief.

(8) Evidence Relied Upon

2003/0050995	MATEOS	03-2003
6,952,737	COATES	10-2005
6,021,437	CHEN	02-2000
6,823,319	LYNCH	11-2004
6,301,590	SIOW	10-2001

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

Claims 29-43, 46, and 47, are rejected under 35 U.S.C. 103(a) as being unpatentable Mateos (US-2003/0050995 03/13/03) in view of Coates (US-6,952,737 10/04/05).

-In regard to independent claim 46, Mateos teaches a server comprising:

a management module configured to generate dynamic data (Paragraph 13: "dynamic information from the server");

a file system storing a web page that had both a first embedded object configured to access the dynamic data and a second embedded object configured to merge the dynamic data with the web page (Paragraphs 29-53: "web pages...example of a web page"; 55: "data section"; 56: "view section"; 58: "the dynamic information is retrieved…display the dynamic information on the client computer"), wherein the first embedded object is executable on a client remote from the server to request the dynamic data (Paragraph 28; Paragraphs 55-59: "browser interprets the HTML tags...a table with the dynamic data information retrieved from the database of the server computer...display of the web page on the client computer is then split into two distinct and consecutive steps…browser is responsible for executing these instructions, in order to display the dynamic information...script implements the Model that holds the dynamic information being manipulated…template implements the View, which manages the graphical and/or textual display of the dynamic information to the user"), and wherein the web page includes a scripting language function defined by the second embedded object, the scripting language function for merging the dynamic data with the web page (Paragraphs 45-53 & 56: "document further includes a view section…result of a document.write JavaScript command");

wherein the server is configured to further:

send, to the client, the web page that has the first embedded object, the second embedded object, and the scripting language function defined by the second embedded object (Paragraphs 29-53, 68-71, and 77-80);

evaluating the scripting language function (Paragraphs 55 & 70);

in response to the request, retrieve the dynamic data and send the retrieved dynamic data to the client for merging with the web page (Paragraphs 29-53, 68-71, and 77-80).

Mateos does not specifically teach wherein the first embedded object executed on the client specifically requests the dynamic data from the server after receiving the webpage from the managed server and evaluating the embedded scripting function. Coates teaches wherein a plurality of SRL objects embedded in a web page executed on a client specifically request dynamic data (e.g. "object files") to be embedded in said web page from the server (column 26, lines 52-67; column 27, lines 1-65)(Figs. 25 & 26) after said web page was received at said client from said server and said SRL object were evaluated (column 26, lines 52-67; column 27, lines 1-65: "generates URL requests to the client site…receives…HTML with one or more embedded SRL(s)…generates SRL requests…file(s) contain content that the client desires to embed in the web page…SRL(s) are embedded in the HTML of the clients web page…with the embedded SRL(s), the end user generates…requests to the storage center…downloads object file to end user"). It would have been obvious to one of ordinary skill in the art at the time of the invention for the dynamic information retrieved in Mateos to have been retrieved from the server based on a request from the client after receiving said HTML web page and after evaluating the embedded objects as taught in Coates, because Coates taught by having the embedded objects of the web

page request the data from the server, the requested object file content gain the benefit of being directly sent to the end-user computer (column 26, lines 52-67; column 27, lines 1-24: “servers object files directly to the end-user computer”).

-In regard to dependent claim 47, Mateos teaches wherein the second embedded object was executable on a client remote from the server to merge the dynamic data with the web page (Paragraph 30: i.e. example web page rendered at the client browser; Paragraph 56: "view section"; Paragraphs 57-60: “browser interprets the HTML tags”).

-In regard to independent claim 37, Mateos teaches a method of displaying a web page, comprising:

requesting at least a frame (Paragraphs 28, 54-56) of a web page (Paragraphs 29-30: “web pages...example of a web page") from a managed server (Paragraph 28: “server…delivers corresponding web pages”), wherein the frame comprised a first embedded object (Paragraphs 34-43 & 55) and a call to a scripting language function defined by the first embedded object (Paragraphs 45-50 & 56), wherein the scripting language function is for merging data corresponding to the first embedded object with the web page (Paragraphs 45-53 & 56: "document further includes a view section…result of a document.write JavaScript command");

receiving the frame from the managed server (Paragraph 56-59);

based on evaluating the scripting language function (Paragraphs 55 & 70), requesting, by the requesting computer, the data (Paragraph 13: “Paragraph 57: “dynamic information”) corresponding to the first embedded object from a managed server after receiving the frame from

the managed server (Paragraphs 56-59: "denote the piece of information to be put in each cell...result of the document.write JavaScript command");

receiving by the requesting computer the data corresponding to the first embedded object (Paragraphs 56-59);

calling by the requesting computer the scripting language function defined by the first embedded object (Paragraphs 45-53 & 56: "document further includes a view section…result of a document.write JavaScript command") and

merging by the requesting computer the data corresponding to the first embedded object into the frame (Paragraph 30: i.e. example web page rendered at the client browser; Paragraph 56: "view section"; Paragraphs 57-60: "browser interprets the HTML tags").

Mateos does not specifically teach wherein the requesting of the data corresponding to the embedded Javascript code was requested from the server after receiving the frame from the managed server and evaluating the embedded scripting function. Coates teaches wherein a plurality of SRL objects embedded in a web page executed on a client specifically request dynamic data (e.g. "object files") to be embedded in said web page from the server (column 26, lines 52-67; column 27, lines 1-65)(Figs. 25 & 26) after said web page was received at said client from said server and said SRL object were evaluated (column 26, lines 52-67; column 27, lines 1-65: "generates URL requests to the client site…receives…HTML with one or more embedded SRL(s)…generates SRL requests…file(s) contain content that the client desires to embed in the web page…SRL(s) are embedded in the HTML of the clients web page…with the embedded SRL(s), the end user generates…requests to the storage center…downloads object file to end user"). It would have been obvious to one of ordinary skill in the art at the time of the invention

for the dynamic information retrieved in Mateos to have been retrieved from the server based on a request from the client after receiving said HTML web page and after evaluating the embedded objects as taught in Coates, because Coates taught by having the embedded objects of the web page request the data from the server, the requested object file content gain the benefit of being directly sent to the end-user computer (column 26, lines 52-67; column 27, lines 1-24: “servers object files directly to the end-user computer”).

-**In regard to dependent claim 38**, Mateos teaches comprises displaying a frame (Paragraph 56-57: "table with the dynamic information retrieved from the database of the server computer is displayed on the client computer").

-In regard to dependent claim 39, Mateos teaches evaluating the frame to identify a source tag of the embedded object (Paragraph 56: "denote the piece of information to be put in each cell...result of the document.write JavaScript command")

-In regard to dependent claim 40, Mateos teaches dynamic data from a management module of the server (Paragraph 29: “retrieve dynamic information requested by the user”; Paragraph 57: “dynamic information retrieved from the database of the server computer”).

-In regard to dependent claim 41, Mateos teaches wherein the dynamic data was generated at run time in response to the request for data corresponding to the embedded object (Paragraph 3: “download and display information whose content changes each time it is provided

by the server computer"; Paragraph 13: "requesting, by the client computer, the dynamic information from a server computer of the network").

-In regard to dependent claim 42, Mateos teaches wherein the data corresponding to the first embedded object comprises a scripting language function (Paragraph 55: "tag identifying a script written in the JavaScript language")

-In regard to dependent claim 43, Mateos teaches wherein the frame could comprises a plurality of embedded objects linked to dynamic data in the managed server, and wherein the scripting language function was configured to merge the dynamic data with the frame (Paragraph 30: i.e. example web page rendered at the client browser; Paragraph 56: "view section"; Paragraphs 57-60: "browser interprets the HTML tags").

-In regard to independent claim 29, Mateos teaches a method for serving data from a managed server, comprising:

serving a web page (Paragraphs 29-30: "web pages...example of a web page") to a requesting computer (Paragraph 13: "requesting, by the client computer") from a managed server (Paragraph 28: "server…delivers corresponding web pages"), the web page comprising a source call (Paragraphs 29-43 & Paragraph 55) to an object file (Paragraphs 29 & 55-56) and code including scripting language functions defined by the object file (Paragraphs 45-50 & 56), wherein the requesting computer was remote from the managed sever (Fig. 2: 125c & 125s), and wherein at least one of the scripting functions is for merging data associated with the object file

with the web page (Paragraphs 56-59: "denote the piece of information to be put in each cell...result of the document.write JavaScript command");

receiving a request from the requesting computer to the managed sever for the object file (Paragraph 57-59),

populating the object file in real-time with data from a management module of the managed server (Paragraph 3: "download and display information whose content changes each time it is provided by the server computer"; Paragraph 13: "requesting, by the client computer, the dynamic information from a server computer of the network");

serving the object file to the requesting computer after populating the object file (Paragraphs 57-59).

Mateos does not specifically teach wherein the first embedded object executed on the client specifically requests the dynamic data from the server after receiving the webpage from the managed server and evaluating the embedded scripting function. Coates teaches wherein a plurality of SRL objects embedded in a web page executed on a client specifically request dynamic data (e.g. "object files") to be embedded in said web page from the server (column 26, lines 52-67; column 27, lines 1-65)(Figs. 25 & 26) after said web page was received at said client from said server and said SRL object were evaluated (column 26, lines 52-67; column 27, lines 1-65: "generates URL requests to the client site…receives…HTML with one or more embedded SRL(s)…generates SRL requests…file(s) contain content that the client desires to embed in the web page…SRL(s) are embedded in the HTML of the clients web page…with the embedded SRL(s), the end user generates…requests to the storage center…downloads object file to end user"). It would have been obvious to one of ordinary skill in the art at the time of the invention

for the dynamic information retrieved in Mateos to have been retrieved from the server based on a request from the client after receiving said HTML web page and after evaluating the embedded objects as taught in Coates, because Coates taught by having the embedded objects of the web page request the data from the server, the requested object file content gain the benefit of being directly sent to the end-user computer (column 26, lines 52-67; column 27, lines 1-24: “servers object files directly to the end-user computer”).

-**In regard to dependent claim 30**, Mateos teaches wherein populating the object file comprises populating the object file with a scripting function (Paragraph 30: i.e. note coded web page; Paragraph 55: "script written in JavaScript").

-In regard to dependent claim 31, Mateos teaches wherein the scripting function populated in the object file was JavaScript (Paragraph 55: "script written in JavaScript").

-In regard to dependent claim 32, Mateos teaches wherein populating the object file comprise populating the object file with an array of data (Paragraph 29: “assigns the value of a record (or field) extracted from the database”; Paragraphs 30-59).

-In regard to dependent claim 33, Mateos teaches wherein populating the object file comprises acquiring real-time data indicative of a current status of a server (Paragraph 3: “download and display information whose content changes each time it is provided by the server

computer"; Paragraph 13: "requesting, by the client computer, the dynamic information from a server computer of the network").

-In regard to dependent claim 34, Mateos teaches wherein populating the object ifle comprises providing a language localization file (Paragraph 60: "includes a field defining the language of the strings"; Paragraphs 63 & 66).

-In regard to dependent claim 35, Mateos teaches serving the web page configured for a client computer which included a CPU, a storage memory, and RAM (Paragraph 23). Mateos does not specifically teach wherein the web page was configured for a handheld or palmtop computer platform. It would have been obvious to one of ordinary skill in the art at the time of the invention for the client computer of Mateos to have been a handheld or palmtop computer for receiving the web pages, because PDA's, palmtops, and handhelds were notoriously well known in the art at the time of the invention as web computing devices that provided the benefit of mobile web accesses. Additionally, Mateos teaches distributing the web page in HTML format (Paragraph 28), which was notoriously well known in the art at the time of the invention to be visualized by hand held or palmtop interfaces for the purpose of determining the layout and style of displayed web content.

-In regard to dependent claim 36, Mateos teaches serving the web page and dynamic data across the Internet (Paragraphs 3-4, 13, & 22: "Internet"). Mateos does not specifically teach when serving the web page, comprised serving a web page across a firewall. It would have

been obvious to one of ordinary skill in the art at the time of the invention for the web page of Mateos to have been sent across a firewall, because it was notoriously well known in the art at the time of the invention that firewalls provided the advantage of increased network security by enforcing restrictions on certain users and data.

Claim 44 is rejected under 35 U.S.C. 103(a) as being unpatentable Mateos (US-2003/0050995 03/13/03) in view of Coates (US-6,952,737 10/04/05) in further view of Chen (US-6,021,437 02/01/00).

-In regard to dependent claim 44, Mateos teaches a desire to provide the generated dynamic content from the server each time the content was requested by the client (Paragraph 3: "download and display information whose content changes each time it is provided by the server computer"; Paragraph 13: "requesting, by the client computer, the dynamic information from a server computer of the network"). Mateos does not specifically teach wherein the data corresponding to the embedded object comprises current time and the dynamic data gathered at the managed server at the current time. Chen teaches wherein a client request for dynamic data of a managed server was requested, gathered, and delivered for display at the current real-time (Abstract; column 2, lines 31-67, column 4, lines 10-19). It would have been obvious to one of ordinary skill in the art at the time of the invention for the dynamic data of Mateos to have been created at a current real-time as disclosed in Chen et al, because Chen et al teach said process provides a simple, effective, and inexpensive to implement way for real-time monitoring of data (column 2, lines 31-38).

Claim 45 is rejected under 35 U.S.C. 103(a) as being unpatentable over Mateos (US-2003/0050995 03/13/03) in view of Coates (US-6,952,737 10/04/05) in further view of Lynch et al (US-6,823,319 11-2004).

-In regard to dependent claim 45, Mateos teaches merging the dynamic data into the zones or areas defined by the scripts in the HTML document (Paragraph 55-60). Mateos does not specifically teach populating a drop-down menu with a dynamic data menu item. Lynch et al teach utilizing static HTML as well as a script to populate a drop-down menu with menu items for display to a user (column 7, lines 1-8). It would have been obvious to one of ordinary skill in the art at the time of the invention for the scripted dynamic content of Mateos to have populated a drop-down menu, because Lynch taught that said method saves time by creating the dynamic data from previously entered data (column 2, lines 8-24; column 7, lines 1-15).

(10) Response to Argument

Appellant's arguments filed 04/30/10 have been fully considered but they are not persuasive.

A: Claims 28-43, 46, and 47

-In regard to claims 37-40, 42, and 43, Appellant argues that neither Matoes nor Coates teaches wherein the request for the data corresponding to the first embedded object from the server was performed, "after receiving the frame of a web page from the managed server, where

the frame has a first embedded object and a call to a scripting language function defined by the first embedded object." The Examiner respectfully disagrees with the Appellant. The Examiner notes that the basic function of the Matoes reference was to provide dynamic information from a server database to a requesting client in the form of a web page (Paragraph 13: "delivering dynamic information in a network"). The best mode by which the Matoes reference achieves this function was by accessing said dynamic information at the server based on a specific client request and sending the retrieved dynamic information along with a web page structure to the client such that the client can render the web page with the embedded dynamic information (Paragraph 13: "requesting by a client computer the dynamic information from a server…retrieving the dynamic information under the control of the server...transmitting the dynamic information and the view template from the server computer to the client computer and combining the dynamic information with the view template under the control of the client computer for display")(Fig. 4). To do this Matoes clearly defines the code of the generated web page (Paragraphs 30-53: "the following code provides a simple example of a web page built from the corresponding view template" & 78: "view template with the inserted statements defines a web page; the instructions are markup tags and statements are script code") to include a scripting language function defined in a HTML frame for requesting the dynamic data to be incorporated into a retrieved web page as well as evaluating a call to said scripting language function to merge said retrieved data with said web page (Paragraph 55: "HTML document has a data section…a tag identifying a script written in the JavaScript language…and the dynamic information resulting from a query on the database"; Paragraph 56: "HTML document further includes a view section…a table is defined…and the result of the document.write Javascript

command"; Paragraph 70-71: "CGI program retrieves that view template….program runs a first query on the database…browser resumes its operation in response to the web page received from the server…according the instructions stored in its view section and using the dynamic information stored in the script variables of its data section").

That being said, the Examiner agrees that the Mateos reference does not specifically teach wherein the web page was first sent to the client and then the first embedded object was evaluated for a corresponding data request and data merge operation (i.e. as shown above, the Matoes reference teaches wherein the scripting language function for retrieving the dynamic data was evaluated and performed at the server based on the client request and wherein the client browser processed the returned HTML page to merge the retrieved data with said page for display on the client computer)(Paragraph 57-58)(Fig. 4). However as noted above in the rejection, the Coates reference clearly teaches wherein a web page with embedded SRL objects was served to a client device and after receiving said web page the embedded objects were evaluated at said client and corresponding data requests and merge operations were performed (column 26, lines 52-67; column 27, lines 1-65: "generates URL requests to the client site…receives…HTML with one or more embedded SRL(s)…generates SRL requests…file(s) contain content that the client desires to embed in the web page…SRL(s) are embedded in the HTML of the clients web page…with the embedded SRL(s), the end user generates…requests to the storage center…downloads object file to end user"). Coates teaches for an end user computer to render a given web page with specific files containing content to be shown in said web page, that a user makes a first URL request to a "content web server" (Fig. 25: 2630) whereby the web server "dynamically generates the SRL(s)" which are then embedded in the

requested HTML page and downloaded by the client. The end user computer then processes the SRL(s) embedded in the downloaded HTML page and directly requests the specific content to be rendered in said web page from a storage center using the SRL(s). The specific content is then downloaded directly to the end user computer and merged with the HTML document for display (Figs. 25 & 26). Coates teaches that it would have been obvious to one of ordinary skill in the art at the time of the invention for the dynamic information retrieved in Mateos to have been retrieved from the server based on a request from the client after receiving said HTML web page at said client and after evaluating the embedded objects at said client, because Coates taught that by having the embedded objects of the web page request the data from the server, the requested object file content gained the benefit of being directly sent to the end-user computer (column 26, lines 52-67; column 27, lines 1-24: "servers object files directly to the end-user computer"). This motivation would have provided a significant potential benefit to the system of Mateos which was concerned with reducing the load on the server and thus overloading its performance (Paragraph 6: "CGI program running the query and building…severely affects the performance of the server computer…increases response time…slowing down"; Paragraphs 10-11: "improve performance of the network…reduce response time of the server computer"; Paragraphs 74-77).

Appellant further argues that the SRLs of Coates do not constitute a scripting language function of as described in claim 37 in that the SRLs do not merge data corresponding to the first embedded object with the web page. The Examiner respectfully disagrees with the Appellant. As noted above, the Mateos reference has been specifically relied upon to teach a call to a scripting language function for requesting and merging data corresponding to the first

embedded object (Paragraph 55: “data section"; Paragraph 56: "HTML document further includes a view section"; Paragraph 70). Thus the Coates reference has not been relied upon to teach the well known functionality of embedded scripts for requesting and merging data but rather has been relied upon to teach embedding objects within a requested page such said objects further request dynamic information after said requested page has been sent to a client. Coates does teach wherein the SRL are embedded objects which were utilized for accessing and merging data with the web page (column 27, lines 11-46: “contain content that the client desires to embed in the web page…downloads the object file to the user”).

Wherein the Appellant argues that validity of the previously mentioned Siow reference, the Examiner notes that said reference was only mentioned in light of Appellant’s previous general arguments. While the Examiner has shown that each of claimed elements are taught in the rejection above, the Siow reference was additionally pointed to for showing that the features to which Appellant has argued to be novel are generally considered well known functionality at the time of the invention.

Additionally, the Appellant also argues that Mateos fails to teach or suggest that the web page of Mateos includes a scripting language function for merging data corresponding to the first embedded object based on an evaluation of the scripting language function. The Examiner respectfully disagrees with the Appellant, and as noted above, the view section of the HTML document of the Mateos reference clearly teaches evaluating a Javascript command corresponding to the first embedded object in the data section of the web page for merging the retrieved data into the web page (Paragraph 56: i.e. the retrieved data was merged into different cells of a table in the returned HTML web page; Paragraph 71).

-In regard to dependent claim 41, Appellant further argues that neither Matoes nor Coates teaches that the dynamic data was generated in real-time in response to the request for the data corresponding to the first embedded object. The Examiner respectfully disagrees. The Matoes reference clearly teaches wherein information requested by the client was dynamic information queried from a server database at the time of the client request in relation to the scripting language function (Paragraph 3: "download and display information whose content changes each time it is provided by the server computer"; Paragraph 13: "requesting, by the client computer, the dynamic information from a server computer of the network"; Paragraph 68: "receive dynamic information from the server computer…database query"; Paragraphs 69-70: "user inputs a filter for the dynamic information...runs a first query on the database according...to the CGI program"). As discussed above in relation to independent claim 37, the Examiner agrees that the Matoes reference does not teach wherein the dynamic data request was made after receiving the requested web page but notes that the Coates reference has been relied upon to teach the specific timing of receiving the web page and then making the request for the dynamic information (column 26, lines 52-67; column 27, lines 1-65: "generates URL requests to the client site…receives…HTML with one or more embedded SRL(s)…generates SRL requests…file(s) contain content that the client desires to embed in the web page…SRL(s) are embedded in the HTML of the clients web page…with the embedded SRL(s), the end user generates…requests to the storage center…downloads object file to end user")(Figs. 25 & 26).

-In regard to claims 29-32, 34-36, 46, and 47, Appellant argues similarly to independent claim 37 in the belief that neither the Mateos nor Coates references teach or suggest wherein the request from the client was received "after the web page has been served to the requesting client computer and after the requesting computer has evaluated the at least one scripting function contained in the web page." The Examiner respectfully disagrees and points to the arguments addressed above with regard to independent claim 37 in addition to the rejections of independent claims 29, 37, and 46 as described above in the Grounds of Rejection section.

-In regard to dependent claim 33, Appellant further argues that the data gathered by the Matoes and/or Coates reference does not constitute real-time data indicative of a current status of a server. The Examiner respectfully disagrees with the Appellant. The Mateos reference clearly teaches wherein the user can request dynamic information from the server (Paragraph 3: "download and display information whose content changes each time it is provided by the server computer"; Paragraph 13: "requesting, by the client computer, the dynamic information from a server computer of the network"; Paragraph 29: "runs queries on a database…client computer" Paragraph 68: "receive dynamic information from the server computer"). Figure 2 shows that the server computer from which the user requests information maintains the database whereby the dynamic information is retrieved. Thus because Matoes teaches that the server could maintain the requested dynamic data, the dynamic data clearly is indicative of a current status of the server. The Examiner notes that what it is for the data to be indicative of the current status of a server is not clearly defined in the claims. Thus one of ordinary skill in the art would

appreciate that if a user requested dynamic data stored at a given server that said returned data would constitute a current status of said server (e.g. that said server still maintains that data, etc).

B: Claim 44

-In regard to dependent claim 44, Appellant argues that that Chen does not teach or suggest wherein the data corresponding to the first embedded object "comprised both a **current time** and dynamic data gathered at the managed server **at the current time**." The Examiner respectfully disagrees with the Appellant. The Chen reference clearly teaches a system that provides real-time monitoring of the status of a server by measuring and storing parameters indicating the status and the behavior of the server at a given moment (Abstract; column 2, lines 31-67, column 4, lines 10-19). When the information of the status of the server is requested from the client, said server writes code in HMTL language such that the information requested can be displayed on the client browser (column 4, lines 13-18). Chen clearly teaches wherein the information written to and displayed in the HTML page included both dynamic data gathered at the server at the current time in addition the current time (column 1, lines 40-44: "up to the present time, providing real-time monitoring"; column 2, lines 42-45: "measuring, and storing parameters…at a given moment"; column 6, lines 20-25, 36-51, & 59-63: "recording the history of events with a date...then display on the client's screen"; column 9, lines 35-43: "at a set date…dates and times for the collection of data"; column 10, lines 4-8: "data collected from various machines…relative to a predefined criteria (date…etc.)"; column 11, lines 23-30: "possible to present general information on the contents of the files: the date"). Chen shows that when observing, recording, and displaying requested dynamic real-time data the information

presented to the user as dynamic web pages included a plurality of different information/parameters to specifically include the date and/or time the information was observed/created/retrieved. Thus part of the data gathered for an event at a current time includes the current time the event occurred.

C: Claim 45

-In regard to dependent claim 45, Appellant argues that due to its dependence on independent claim 37, claim 45 is allowable for similar reasons as previously cited. The Examiner respectfully disagrees and points to the arguments addressed above with regard to independent claim 37 in addition to the rejection of both claims 37 and 45 as described above in the Grounds of Rejection section.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/Adam L Basehoar/

Primary Examiner, Art Unit 2178

07/18/10

Conferees:

/Stephen S. Hong/

Supervisory Patent Examiner, Art Unit 2178

Stephen Hong, Supervisory Patent Examiner AU 2178

/William L. Bashore/

Supervisory Patent Examiner, Art Unit 2175

William Bashore, Supervisory Patent Examiner AU 2175